Model-based Clustering of Time Series in Group-specific Functional Subspaces

Charles Bouveyron[1] & Julien Jacques[2]

[1] Laboratoire SAMM, EA 4543
Université Paris 1 Panthéon-Sorbonne

[2] Laboratoire Paul Painlevé, UMR CNRS 8524
Université Lille 1 & INRIA Lille-Nord Europe

Abstract

This work develops a general procedure for clustering functional data which adapts the clustering method High Dimensional Data Clustering (HDDC), originally proposed in the multivariate context. The resulting clustering method, called funHDDC, is based on a functional latent mixture model which fits the functional data in group-specific functional subspaces. By constraining model parameters within and between groups, a family of parsimonious models is exhibited which allow to fit onto various situations. An estimation procedure based on the EM algorithm is proposed for determinating both the model parameters and the group-specific functional subspaces. Experiments on real-world datasets show that the proposed approach performs better or similarly than classical two-step clustering methods while providing useful interpretations of the groups and avoiding the uneasy choice of the discretization technique. In particular, funHDDC appears to always outperform HDDC applied on spline coefficients.

1 Introduction

Cluster analysis consists in identifying groups of homogeneous data without using any prior knowledge on the group labels of the data. A lot of methods, from k-means (Hartigan and Wong, 1978) or hierarchical classification to more recent probabilistic model-based clustering (Banfield and Raftery, 1993; Celeux and Govaert, 1995), have been proposed along the years. The clustering of time series, or more generally of functions, is a difficult task since the data live in an infinite dimensional space. We refer for instance to Warren Liao (2005) for a survey on time series clustering. Although non-parametric approaches to functional clustering, as

for instance Ferraty and Vieu (2006); Tarpey and Kinateder (2003), lead to powerful clustering algorithms, the present paper focuses on model-based clustering which has moreover interesting interpretability properties.

Unlike in the case of finite dimensional data vectors, model-based methods for clustering functional data are not directly available since the notion of probability density function generally does not exist for such data (Delaigle and Hall, 2010). Consequently, the use of model-based clustering methods on functional data consists usually in first transforming the infinite dimensional problem into a finite dimensional one and then in using a model-based clustering method designed for finite dimensional data. The representation of functions in a finite space can be carried out by either discretizing the time interval, decomposing the functions in a basis of functions or by means of some principal components resulting from a functional principal component analysis (FPCA, Ramsay and Silverman (2005)). The discretization of the time interval is usually straightforward since in practice the functions are already measured in a discrete scale. On the other hand, the choice of a basis of functions may include well-defined functions such as natural cubic splines which are very popular and enjoy some optimality properties (Wahba, 1990). Alternatively, the decomposition of the functions can be done through specific time series models such as ARMA or GARCH (see Frühwirth-Schnatter and Kaufmann (2008) for a clustering algorithm based on such models). Note finally that in the case of using functional principal components, the data functions have also to be decomposed in a basis of functions in order to solve the functional eigen-decomposition problem.

Unfortunately, the resulting vectors are often high-dimensional. In particular, the discretization of the observed curves usually leads to high-dimensional datasets with sometimes less observations than dimensions. In such situations, model-based clustering methods suffer from numerical problems and regularized approaches have to be used. Among the regularized model-based clustering methods, we can cite the parsimonious Gaussian mixture models (Banfield and Raftery, 1993; Celeux and Govaert, 1995), which assume specific covariance structures, mixture of probabilistic principal component analyzers (MixtPPCA, Tipping and Bishop (1999)) and high-dimensional data clustering (HDDC, Bouveyron et al (2007)) which both assume that high-dimensional data live in group-specific subspaces. In particular, the latter method has been used successfully in various application fields such as image analysis (Bouveyron et al, 2007) or chemometry (Jacques et al, 2010).

The two-step approaches previously described perform the discretization and the clustering steps separately, and this may lead to a loss of discriminative information. Recently, a new approach due to James and Sugar (2003) allows the interaction between the discretization and the clustering steps by introducing a stochastic model for the basis coefficients. This approach is deemed by its authors to be particularly effective when the functional data are sparsely sampled. In a similar spirit, we propose in the present paper to adapt the HDDC method to functional data in order to model and cluster the functional data in group-specific subspaces of low dimensionality. The modeling of the functions of each group in a specific subspace should, in addition to providing an eventually interesting clustering of the data,

facilitate the interpretation of the clustered data.

The paper is organized as follows. Section 2 presents the proposed functional latent mixture model as well as a family of parsimonious submodels. Maximum likelihood estimation is described in Section 3. Section 4 first proposes an introductory example in order to highlight the main features of the proposed method. A benchmark comparison with state-of-the-art methods is also provided in Section 4 on real-world time series datasets. Finally, Section 5 provides some concluding remarks.

2 The functional latent mixture model

Clustering of functional data consists in identifying K homogeneous groups (or clusters) among the data at hand (curves, times series or functions). To do this in a model-based context, this section introduces a family of mixture models designed for functional data which adapts the models of (Bouveyron et al, 2007) proposed in the multivariate context.

2.1 Transformation of the observed curves

Let us first assume that the observed curves $\{x_1, ..., x_n\}$ are independent realizations of a L_2-continuous stochastic process $X = \{X(t)\}_{t\in[0,T]}$ for which the sample paths, *i.e.* the observed curves, belong to $L_2[0, T]$. In practice, the functional expressions of the observed curves are not known and we only have access to the discrete observations $x_{ij} = x_i(t_{ij})$ at a finite set of ordered times $\{t_{ij} : j = 1, \ldots, m_i\}$. As explained in Aguilera et al (2011), it is therefore necessary to first reconstruct the functional form of the data from their discrete observations. A common way to do this is to assume that curves belong to a finite dimensional space spanned by a basis of functions (see for example Ramsay and Silverman (2005)). Let us therefore consider such a basis $\{\psi_1, \ldots, \psi_p\}$ and assume that the stochastic process X admits the following basis expansion:

$$X(t) = \sum_{j=1}^{p} \gamma_j(X)\psi_j(t), \tag{1}$$

where $\gamma = (\gamma_1(X), ..., \gamma_p(X))$ is a random vector in $\mathbb{R}^p$ and the number p of basis functions is assumed to be fixed and known. The basis expansion of each observed curve $x_i(t) = \sum_{j=1}^{p} \gamma_{ij}\psi_j(t)$ can be estimated by an interpolation procedure (see Escabias et al (2005) for instance), if the curves are observed without noise, or by least square smoothing, if they are observed with error. In the present paper the second option will be used. In the following, a latent mixture model is proposed for the modeling of the coefficient vectors $\{\gamma_1, ..., \gamma_n\} \in \mathbb{R}^p$ of the observed curves $\{x_1, ..., x_n\}$ that one wants to cluster, where $\gamma_i = (\gamma_{i1}, \ldots, \gamma_{ip})$ for $i = 1, ..., n$.

2.2 A group-specific functional latent model

Let us now consider just a set of n_k observed curves, described by their coefficient vectors $\{\gamma_1, ..., \gamma_{n_k}\} \in \mathbb{R}^p$, belonging to the same cluster, the kth cluster. Let us first assume that $\{\gamma_1, ..., \gamma_{n_k}\}$ are independent realizations of a random vector $\Gamma \in \mathbb{R}^p$. Let us also assume that the actual stochastic process associated with the kth cluster can be described in an adequate way in a low-dimensional functional latent subspace $\mathbb{E}_k[0,T]$ of $L_2[0,T]$ with dimension $d_k \leq p$. Let $\mathbb{E}_k[0,T]$ be spanned by the first d_k elements of a group-specific basis of functions $\{\varphi_{kj}\}_{j=1,...,d_k}$ in $L_2[0,T]$. This group-specific basis is obtained from $\{\psi_j\}_{j=1,...,p}$ by a linear transformation $\varphi_{kj} = \sum_{\ell=1}^{p} q_{k,j\ell}\psi_\ell$ with an orthogonal $p \times p$ matrix $Q_k = (q_{k,j\ell}) = [U_k, V_k]$ that is split, for later use, into two parts: U_k of size $p \times d_k$ and V_k of size $p \times (p - d_k)$ with $U_k^t U_k = I_{d_k}$, $V_k^t V_k = I_{p-d_k}$ and $U_k^t V_k = 0$.

Let $\{\lambda_1, ..., \lambda_{n_k}\}$ be the latent expansion coefficients of the curves in the group-specific basis $\{\varphi_{kj}\}_{j=1,...,d_k}$. These coefficients are also assumed to be independent realizations of a latent random vector $\Lambda \in \mathbb{R}^{d_k}$. The relationship between both bases $\{\varphi_{kj}\}_{j=1,...,d_k}$ and $\{\psi_j\}_{j=1,...,p}$ suggests that the random vectors Γ and Λ are linked through the following linear transformation for the kth group:

$$\Gamma = U_k \Lambda + \varepsilon, \tag{2}$$

where $\varepsilon \in \mathbb{R}^p$ is an independent and random noise term.

We now make some distributional assumptions on the random vectors Λ and ε. Firstly, Λ is assumed to be distributed according to a multivariate Gaussian density:

$$\Lambda \sim \mathcal{N}(m_k, S_k), \tag{3}$$

where m_k and $S_k = \mathrm{diag}(a_{k1}, ..., a_{kd_k})$ are respectively the mean and the covariance matrix of the kth group. Secondly, ε is assumed to be distributed according to a multivariate Gaussian density:

$$\varepsilon \sim \mathcal{N}(0, \Xi_k). \tag{4}$$

With these distributional assumptions, the distribution of Γ for the kth cluster is finally:

$$\Gamma \sim \mathcal{N}(\mu_k, \Sigma_k), \tag{5}$$

where $\mu_k = U_k m_k$ and $\Sigma_k = U_k S_k U_k^t + \Xi_k$.

We finally assume that the noise covariance matrix Ξ_k is such that $\Delta_k = \mathrm{cov}(Q_k^t \Gamma) = Q_k^t \Sigma_k Q_k$ has the following form:

$$\Delta_k = \begin{pmatrix} \boxed{\begin{matrix} a_{k1} & & 0 \\ & \ddots & \\ 0 & & a_{kd_k} \end{matrix}} & \mathbf{0} \\ \mathbf{0} & \boxed{\begin{matrix} b_k & & 0 \\ & \ddots & \\ & & \ddots \\ 0 & & b_k \end{matrix}} \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} d_k \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} (p - d_k) \end{matrix} \tag{6}$$

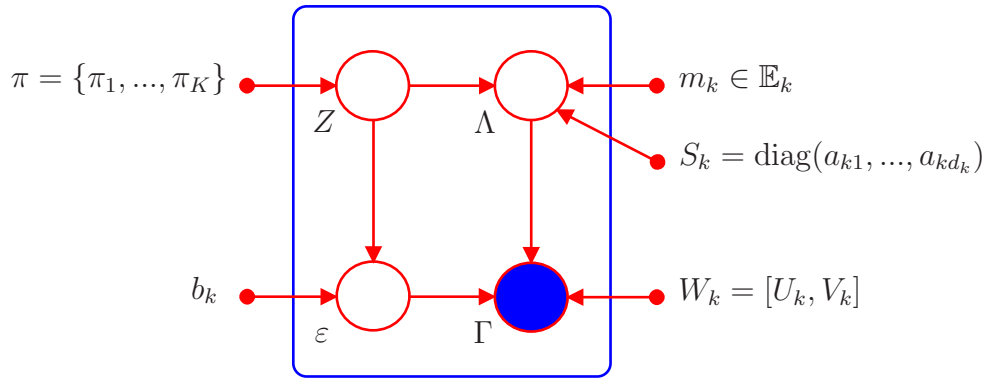


Figure 1: Graphical summary of the functional latent mixture (FLM) model.

with $a_{kj} > b_k$ for $j = 1, ..., d_k$. With these notations and from a practical point of view, one can say that the variance of the actual data of the kth group is therefore modeled by $a_{k1}, ..., a_{kd_k}$ whereas the parameter b_k models the variance of the noise. Similarly, the dimension d_k can be considered as the intrinsic dimension of the latent subspace of the kth group. Figure 1 summarizes all these notations.

2.3 The functional latent mixture model and its submodels

Let us now consider a set of n observed time series or curves $\{x_1, ..., x_n\}$, where $x_i = \{x_i(t)\}_{t\in[0,T]}$ $(1 \leq i \leq n)$, that one wants to cluster into K homogeneous groups. Let us assume that there exists an unobserved random variable $Z = (Z_1, \ldots, Z_K) \in \{0, 1\}^K$ indicating the group membership of X: Z_k is equal to 1 if X belongs to the kth group and 0 otherwise. The clustering task aims therefore to predict the value $z_i = (z_{i1}, \ldots, z_{iK})$ of Z for each observed curve x_i.

As previously, each curve x_i is assumed to be a sample path of X, admitting a basis expansion summarized by the coefficient vector γ_i whose distribution is now a mixture of Gaussians with density:

$$p(\gamma) = \sum_{k=1}^{K} \pi_k \phi(\gamma; \mu_k, \Sigma_k), \tag{7}$$

where ϕ is the standard Gaussian density function, $\mu_k = U_k m_k$, $\Sigma_k = Q_k \Delta_k Q_k^t$ and $\pi_k = P(Z_k = 1)$ is the prior probability of the kth group. This mixture model will be hereafter referred to as the $\text{FLM}_{[a_{kj}b_kQ_kd_k]}$ model or the FLM (Functional Latent Mixture) model for short.

Following the strategy of Bouveyron et al (2007), it is possible to obtain parsimonious submodels from the $\text{FLM}_{[a_{kj}b_kQ_kd_k]}$ model by constraining model parameters within or between groups. For instance, fixing the first d_k diagonal elements of Δ_k to be common within each class, we obtain the restricted model $\text{FLM}_{[a_kb_kQ_kd_k]}$. We observed that the model $\text{FLM}_{[a_kb_kQ_kd_k]}$ often gives satisfying results (in term of clustering accuracy) and this suggests that the assumption that each matrix Δ_k contains

FLM model	Number of parameters	Nb of prms $K = 4$, $d_k = 10$, $p = 100$
$[a_{kj}b_kQ_kd_k]$	$\rho + \tau + 2K + D$	4231
$[a_{kj}bQ_kd_k]$	$\rho + \tau + K + D + 1$	4228
$[a_kb_kQ_kd_k]$	$\rho + \tau + 3K$	4195
$[ab_kQ_kd_k]$	$\rho + \tau + 2K + 1$	4192
$[a_kbQ_kd_k]$	$\rho + \tau + 2K + 1$	4192
$[abQ_kd_k]$	$\rho + \tau + K + 2$	4189

Table 1: Properties of the FLM models: $\rho = Kp+K-1$ is the number of parameters required for the estimation of means and proportions, $\tau = \sum_{k=1}^{K} d_k[p - (d_k + 1)/2]$ are the number of parameters required for the estimation of orientation matrices Q_k, and $D = \sum_{k=1}^{K} d_k$.

only two different eigenvalues, a_k and b_k, seems to be an efficient way to regularize the estimation of Δ_k. Another possible type of regularization is to fix the parameters b_k to be common between the classes. This yields the models $\text{FLM}_{[a_{kj}bQ_kd_k]}$ and $\text{FLM}_{[a_kbQ_kd_k]}$ which both assume that the behavior of the error components outside the class specific subspaces is common. This assumption can be viewed as modeling the noise outside the latent subspace of the group by a single parameter b which is a natural hypothesis when the data are obtained in a common acquisition process. Among the 28 models proposed in the original article (Bouveyron et al, 2007), 6 models have been selected for their good practical behaviors to be considered in the experiments of Section 4. Table 1 lists those 6 models and their corresponding complexity (*i.e.* the number of parameters to estimate).

2.4 Links with related models

At this point, it is possible to establish some links with the methods mentioned in Section 1. The closest strategy is obviously the direct use of HDDC on the basis coefficients. This implies that a "standard" PCA is applied, conditionally on the group membership posterior probabilities, to the data of each group. The main difference between HDDC (Bouveyron et al, 2007) and its functional version, described in Sections 2.2 and 2.3, is the use of a metric specific to the functional data in the eigenspace projection. It is also possible to directly use HDDC on the discretized data. In this case, the functional nature of the data is not considered at all, what could be especially problematic when the curves are observed with noise. The experiments presented in Section 4 will show that the use of the functional version of HDDC both improve the clustering results and facilitate the interpretation of the results by looking at the group-specific functional harmonics.

3 Model inference: the funHDDC algorithm

In model-based clustering, the estimation of model parameters is traditionally done by maximizing the likelihood through the EM algorithm (Dempster et al, 1977). This iterative algorithm consists in maximizing the complete likelihood rather than directly maximizing the likelihood which is an intractable problem with incomplete data (the cluster memberships are unknown here). This section presents the update formula of the EM algorithm in the case of the FLM model.

3.1 FunHDDC: an EM-based algorithm

Given the coefficient vectors $\gamma_1, ..., \gamma_n$ of the observed curves $x_1, ..., x_n$, the complete log-likelihood of the data under the FLM model proposed above has the following form:

$$\begin{aligned}\ell_c(\theta; \gamma_1, ..., \gamma_n, z_1, ..., z_n) &= -\frac{1}{2}\sum_{k=1}^{K}\eta_k\left[\sum_{j=1}^{d_k}\left(\log(a_{kj}) + \frac{q_{kj}^t C_k q_{kj}}{a_{kj}}\right)\right. \\ &\quad \left. + \sum_{j=d_k+1}^{p}\left(\log(b_k) + \frac{q_{kj}^t C_k q_{kj}}{b_k}\right) - 2\log(\pi_k)\right] + \xi, \quad (8)\end{aligned}$$

where $\theta = (\pi_k, \mu_k, a_{kj}, b_k, q_{kj})$ for $1 \leq j \leq d_k$ and $1 \leq k \leq K$, q_{kj} is the jth column of Q_k, $C_k = \frac{1}{\eta_k}\sum_{i=1}^{n} z_{ik}(\gamma_i - \mu_k)^t(\gamma_i - \mu_k)$, $\eta_k = \sum_{i=1}^{n} z_{ik}$ and ξ is a term not depending on the parameter θ. As the class memberships z_{ik} are unknown, it is necessary to estimate them (E step) before to maximize the complete likelihood (M step). These two steps of the EM algorithm are described in details in the following.

E step This first step aims to compute, at iteration q, the expectation of the complete log-likelihood conditionally on the current value of the parameter $\theta^{(q-1)}$, which reduces to the computation of $t_{ik}^{(q)} = E[Z_{ik}|\gamma_i, \theta^{(q-1)}]$. For the $\text{FLM}_{[a_k b_k Q_k d_k]}$ model, the posterior probability $t_{ik}^{(q)}$ can be computed as follows at iteration q:

$$t_{ik}^{(q)} = 1/\sum_{\ell=1}^{K}\exp\left(H_k^{(q-1)}(\gamma_i) - H_\ell^{(q-1)}(\gamma_i)\right), \quad (9)$$

with $H_k^{(q-1)}(\gamma)$ defined for $\gamma \in \mathbb{R}^p$ as:

$$\begin{aligned}H_k^{(q-1)}(\gamma) &= ||\mu_k^{(q-1)} - P_k(\gamma)||_{D_k}^2 + \frac{1}{b_k^{(q-1)}}||\gamma - P_k(\gamma)||^2 \\ &\quad + \sum_{j=1}^{d_k}\log(a_{kj}^{(q-1)}) + (p - d_k)\log(b_k^{(q-1)}) - 2\log(\pi_k^{(q-1)}), \quad (10)\end{aligned}$$

where $||.||_{\mathcal{D}_k}^2$ is a norm on the latent space $\mathbb{E}_k$ defined by $||y||_{\mathcal{D}_k}^2 = y^t\mathcal{D}_k y$, $\mathcal{D}_k = \tilde{Q}\Delta_k^{-1}\tilde{Q}^t$ and $\tilde{Q}$ is a $p \times p$ matrix containing the d_k vectors of U_k completed by zeros

such as $\tilde{Q} = [U_k, 0_{p-d}]$, P_k is the projection operator on the latent space $\mathbb{E}_k$defined by $P_k(\gamma) = U_k U_k^t(\gamma - \mu_k) + \mu_k$.

Let us note that $H_k(\gamma)$ is mainly based on two distances: the distance between the projection of γ on $\mathbb{E}_k$ and the current mean of the kth group and the distance between the observation and the subspace $\mathbb{E}_k$. This classification function favors the assignment of a new observation to the class for which it is close to the subspace and for which its projection on the class subspace is close to the mean of the class. The variance terms a_k and b_k balance the importance of both distances. For example, if the data are very noisy, *i.e.* b_k is large, it is natural to balance the distance $\|\gamma - P_k(\gamma)\|^2$ by $1/b_k$ in order to take into account the large variance outside $\mathbb{E}_k$.

M step This second step estimates the model parameters by maximizing the expectation of the complete likelihood conditionally on the posterior probabilities $t_{ik}^{(q)}$ computed in the previous step. Mixture proportions and means are updated as usually by:

$$\pi_k^{(q)} = \frac{n_k^{(q)}}{n}, \;\; \mu_k^{(q)} = \frac{1}{n_k^{(q)}} \sum_{i=1}^{n} t_{ik}^{(q)} \gamma_i, \tag{11}$$

where $n_k^{(q)} = \sum_{i=1}^{n} t_{ik}^{(q)}$. Let us also introduce $C_k^{(q)} = \frac{1}{n_k^{(q)}} \sum_{i=1}^{n} t_{ik}^{(q)} (\gamma_i - \mu_k^{(q)})^t (\gamma_i - \mu_k^{(q)})$, the sample covariance matrix of group k, and W, the matrix of inner products between the basis functions: $W = (w_{jk})_{1 \leq j,k \leq p} = \int_0^T \psi_j(t)\psi_k(t)dt$. With these notations, the update formula for the other model parameters a_{kj}, b_k and q_{kj} are in the case of the $\text{FLM}_{[a_k b_k Q_k d_k]}$ model, for $k = 1, ..., K$:

- the d_k first columns of Q_k are updated by the eigenvectors associated with the largest eigenvalues of $W^{\frac{1}{2}} C_k^{(q)} W^{\frac{1}{2}}$,
- the variance parameters a_{kj}, $j = 1, ..., d_k$, are updated by the d_k largest eigenvalues of $W^{\frac{1}{2}} C_k^{(q)} W^{\frac{1}{2}}$,
- the variance parameters b_k are updated by $b_k^{(q)} = \text{trace}(W^{\frac{1}{2}} C_k^{(q)} W^{\frac{1}{2}}) - \sum_{j=1}^{d_k} \hat{a}_{kj}^{(q)}$.

Proof of these results can be deduced from the proof of Bouveyron et al (2007), by substituting the usual metric by the metric induced by the basis functions (W here). The inference algorithm presented here will be referred to as funHDDC in the following.

To summarize and roughly speaking, the funHDDC algorithm models and clusters the time series through their projections in group-specific functional principal subspaces. These group-specific functional principal subspaces are obtained by performing functional principal component analysis (Ramsay and Silverman, 2005) conditionally on the posterior probabilities t_{ik}. However, it is important to notice that, even though the modeling and the clustering are conducted in low-dimensional subspace, no discriminative information is lost thanks to the noise term b_k which models the variance outside the subspaces.

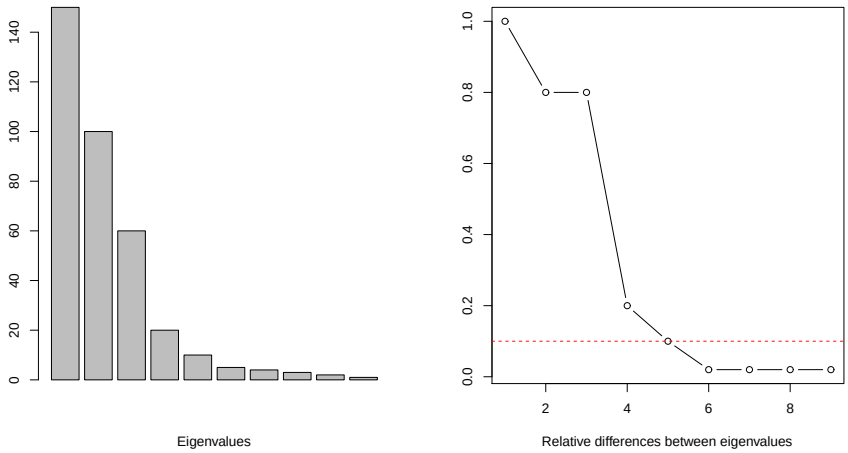


Figure 2: Estimation of the intrinsic dimension of the kth group using the scree test of Cattell. On this example, the intrinsic dimension of the group subspace is estimated to 4.

3.2 Estimation of hyper-parameters

The use of the EM algorithm for parameter estimation makes the funHDDC algorithm almost automatic, except for the estimation of the hyper-parameters d_k and K. Indeed, the parameters d_k and K cannot be determined by maximizing the likelihood since they both control the model complexity. The estimation of the intrinsic dimensions d_k is a difficult problem with no unique technique to use. In (Bouveyron et al, 2007), the authors proposed a strategy based on the eigenvalues of the class conditional covariance matrix Σ_k of the kth class. The jth eigenvalue of Σ_k corresponds to the fraction of the full variance carried by the jth eigenvector of Σ_k. The class specific dimension d_k, $k = 1, ..., K$ is estimated through the scree-test of Cattell (Cattell, 1966) which looks for a break in the eigenvalues scree. The selected dimension is the one for which the subsequent eigenvalues differences are smaller than a threshold. Figure 2 illustrates the use of the Cattell's scree-test. The threshold can be provided by the user or selected using BIC (Schwarz, 1978). The number of clusters K may have to be estimated as well and can be selected by the BIC criterion.

3.3 Classification step

The funHDDC algorithm proposed above aims in the first place to infer the FLM model introduced in the previous section. However, since we are interested in this work in obtaining a partition of the data at hand, it is necessary to add a classification step at the end of the funHDDC algorithm to provide the expected clustering. In the model-based clustering framework, observations are usually assigned to a group using the maximum a posteriori (MAP) rule. The MAP rule assigns an ob-

servation $\gamma_i \in \mathbb{R}^p$ to the group for which γ_i has the highest posterior probability $P(Z_{ik} = 1|\gamma_i)$ at the end of the algorithm. Therefore, this final classification step mainly consists in assigning the observation γ_i to the group with the highest $t_{ik}^{(q_f)}$, $k = 1, ..., K$, where q_f is the last iteration of the algorithm before its convergence.

3.4 Convergence and numerical considerations

Firstly, since the funHDDC algorithm is an EM-based algorithm which respects the classical conditions of the EM theory, its convergence to a local maximum of the likelihood is guaranteed. Several strategies have been proposed in the literature for initializing the EM algorithm in order to avoid the convergence to a local maximum. A popular practice (Biernacki, 2004) executes the EM algorithm several times from a random initialization and keep only the set of parameters associated with the highest likelihood. This initialization procedure is used in the experiments presented in the following section.

Secondly, it is important to remark in Equation (10) that, using the FLM model, the E step does not require as usually to invert the empirical covariance matrices thanks to the form of the matrix Δ. This allows the method to work even in high-dimensional spaces where empirical covariance matrices are usually ill-conditioned. Notice that, since Equation (10) does not require the computation of the $(p - d_k)$ last eigenvectors of Σ_k, funHDDC can also be used when the number of observations per group n_k is smaller than p, as long as $n_k \geq d_k$ for $k = 1, ..., K$.

4 Experimental results

This section presents the results of experiments which aim to both illustrate the funHDDC features and compare the proposed method to existing approaches.

4.1 An introductory example: the Canadian temperature dataset

In this first experiment, the Canadian temperature data (available in the **R** package *fda* and presented in detail in Ramsay and Silverman (2005)) are used to illustrate the main features of the proposed functional clustering method. The dataset consists in the daily measured temperatures at 35 Canadian weather stations across the country. The funHDDC algorithm was applied here with the $[a_{kj}b_kQ_kd_k]$ model, which is the most general FLM model, using a basis of 20 natural cubic splines. Spline functions are smooth piecewise polynomial functions where *cubic* indicates the degree of the polynoms and *natural* indicates the boundary condition (nullity of the second derivative) which ensures the unicity of the spline definition. Refer for instance to Ramsay and Silverman (2005) for more details. Once the funHDDC algorithm has converged, various informations are available and some of them are of particular interest. Group means, intrinsic dimensions of the group-specific sub-

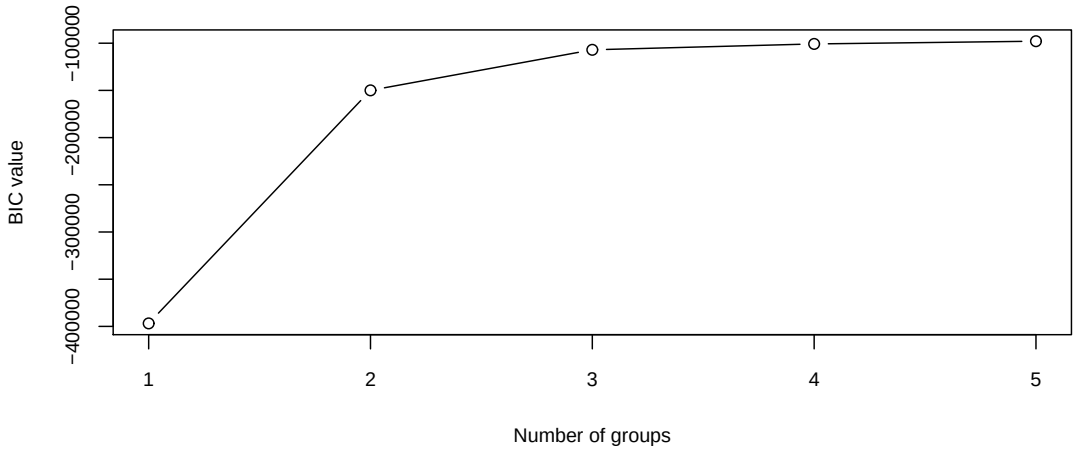


Figure 3: Selection of the number K of groups with the BIC criterion for the Canadian temperature dataset.

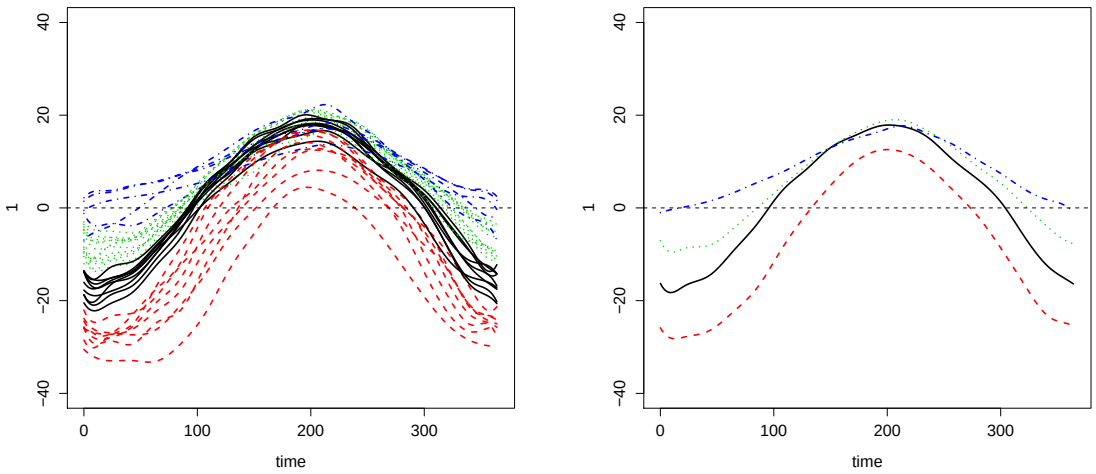


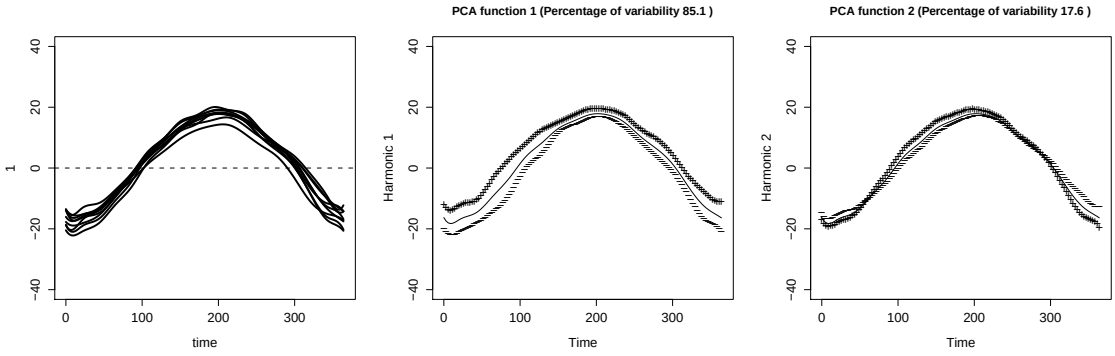
Figure 4: Clustering of the 35 times series obtained with funHDDC (model $[a_{kj}b_kQ_kd_k]$) and estimated mean functions of the groups for the Canadian temperature dataset (see text for details).

Figure 5: Geographical positions of the Canadian weather stations according to their group belonging provided by funHDDC. The colors indicate the group memberships: group 1 in black, group 2 in red, group 3 in green and group 4 in blue.

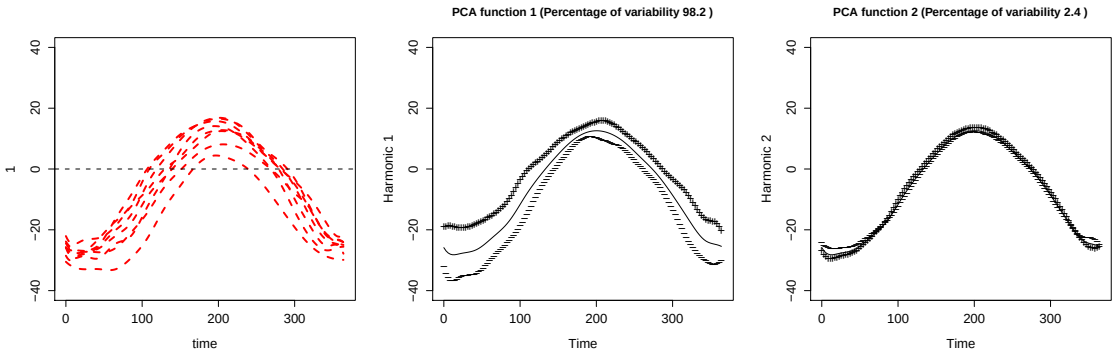
spaces and functional principal components of each group could in particular help the practitioner in understanding the clustering of the dataset at hand.

As discussed before, it is first important to select an appropriate number of components for the dataset to cluster and this can be done using the BIC criterion. Figure 3 shows the BIC values obtained with funHDDC on the Canadian temperature dataset according to the number K of groups. As one can observe, the BIC value increases until $K = 4$ and then stabilizes. Given the behavior of BIC, we decide to select 4 groups rather than higher numbers, even the BIC criterion can be slightly better, in order to facilitate the group interpretation. Figure 4 presents the clustering into 4 groups obtained with funHDDC for the temperature dataset and the estimated mean functions $\bar{x}_k(t) = \sum_{j=1}^{p} \hat{\mu}_{kj}\psi_j(t)$ of the groups where $\hat{\mu}_{kj}$ are the components of $\hat{\mu}_k$ estimated in Section 3.

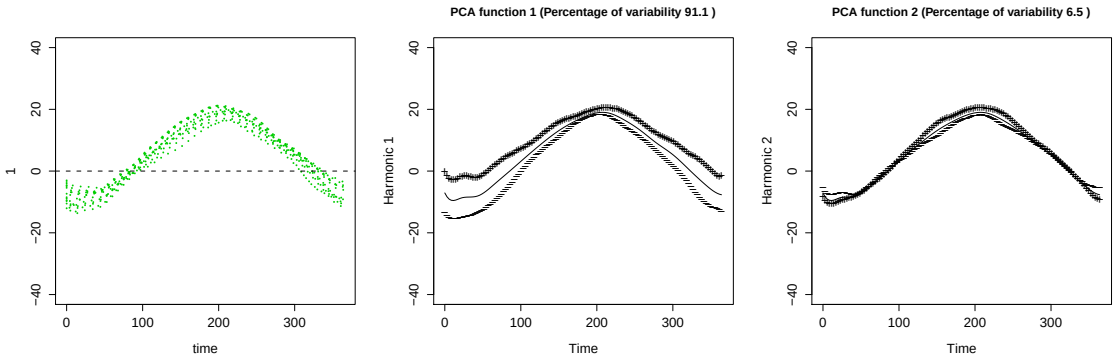
At this point, it is very interesting to have a look at the name of the weather stations gathered in the different groups formed by funHDDC. Indeed, it appears that group 1 (black solid curves) is mostly made of continental stations, group 2 (red dashed curves) mostly gathers the stations of the North of Canada, group 3 (green dotted curves) mostly contains the stations of the Atlantic coast whereas the Pacific stations are mostly gathered in group 4 (blue dot-dashed curves). For instance, group 3 contains stations such as Halifax (Nova Scotia) and St Johns (Newfoundland) whereas group 4 has stations such as Vancouver and Victoria (both in British Columbia). Figure 5 provides a map of the weather stations where the col-



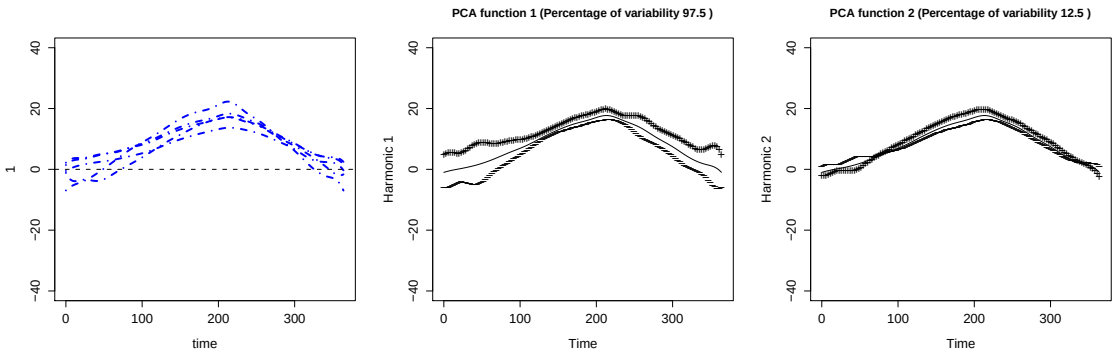
(a) Group 1 (mostly continental stations)



(b) Group 2 (mostly Arctic stations)



(c) Group 3 (mostly Atlantic stations)



(d) Group 4 (mostly Pacific stations)

Figure 6: The group means of the Canadian temperature data obtained with funHDDC and the effects of adding (+) and subtracting (−) twice the square root of the principal component variance (see text for details).

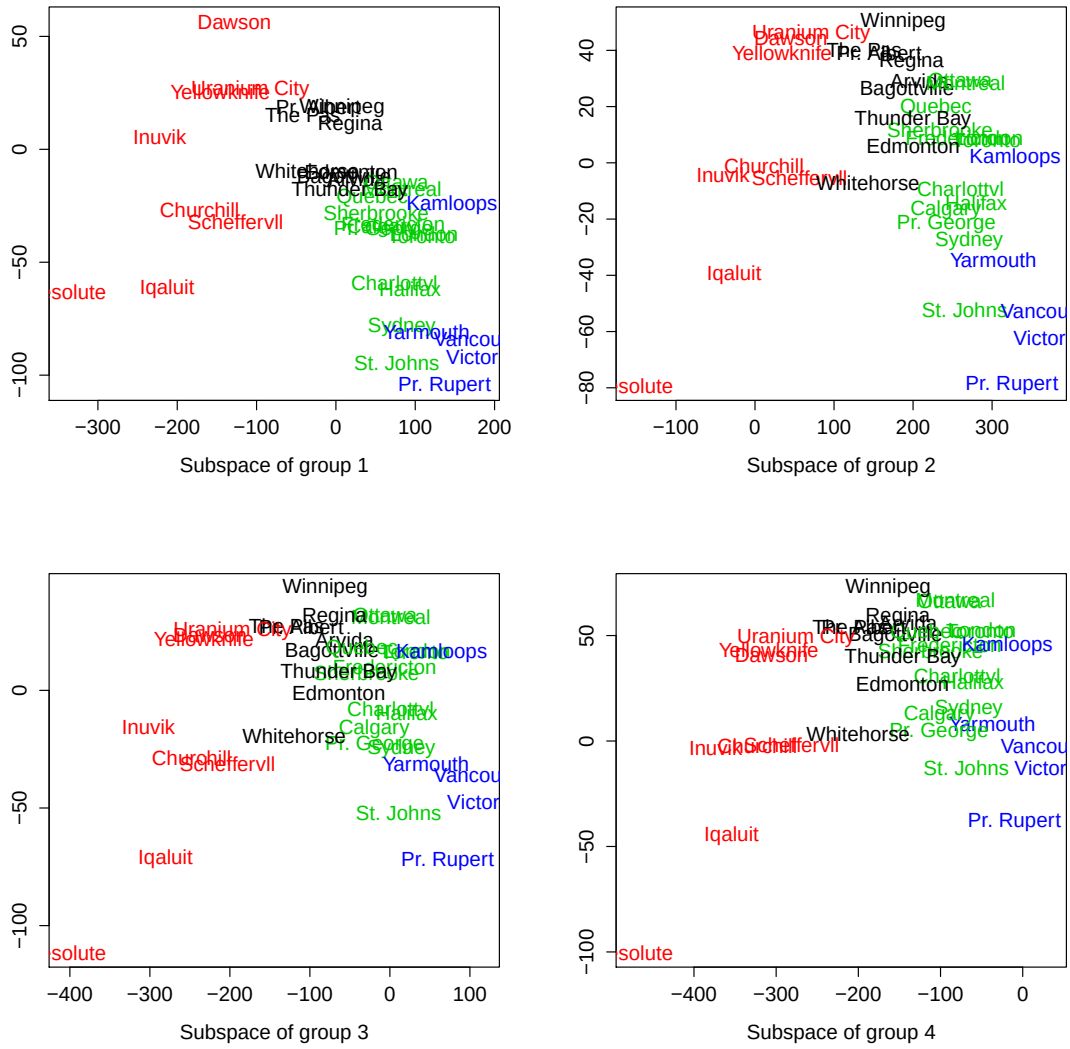


Figure 7: Scores of the observed Canadian temperature curves into the group-specific functional subspaces estimated with funHDDC. The colors indicate the group memberships: group 1 in black, group 2 in red, group 3 in green and group 4 in blue.

ors indicate their group membership. This figure shows that the obtained clustering with funHDDC is very satisfying and rather coherent with the actual geographical positions of the stations (the clustering accuracy is 71% here compared with the geographical classification provided by (Ramsay and Silverman, 2005)). We recall that the geographical positions of the stations has not been used by funHDDC to provide the partition into 4 groups. In addition, the behavior of the temperature means of the 4 groups confirms the common idea that seasons are more rude in the North of Canada than in the South and that the continental cities have lower temperatures than coast cities during the winter.

Another interesting thing, but not necessarily easy to visualize, is the specific functional subspace of each group. A classical way to observe principal component functions is to plot the group mean function $\bar{x}_k$ as well as the functions obtained by adding and subtracting to the mean function twice the square root of the principal component variance, *i.e.* $\bar{x}_k(t) \pm 2\sqrt{a_{kj}}$ for the jth principal component of group k. Refer to (Ramsay and Silverman, 2005) for more details on this usual representation. Figure 6 shows such a plot for the 4 groups of weather stations formed by funHDDC. It first appears on the first principal component of each group that there is more variance between the weather stations in winter than in summer. In particular, the first principal component of group 4 (blue curves, mostly Pacific stations) reveals a specific phenomenon which occurs at the beginning and the end of the winter. Indeed, we can observe a high variance in the temperatures of the Pacific coast stations at these periods of time which can be explained by the presence of mountain stations in this group. The analysis of the second principal components reveals more fine phenomena. For instance, the second principal component of group 1 (black curves, mostly continental stations) shows a slight shift between the + and − along the year which indicates a time-shift effect. This may mean that some cities of this group have their seasons shifted, *e.g.* late entry and exit in the winter. Similarly, the inversion of the + and − on the second principal component of the Pacific and Atlantic groups (blue and green curves) suggests that, for these groups, the coldest cities in winter are also the warmest cities in summer. On the second principal component of group 2 (red curves, mostly Arctic stations), the fact that the + and − curves are almost superimposed shows that the North stations have very similar temperature variations (different temperature means but same amplitude) along the year.

Finally, Figure 7 presents the scores of the curves into the two first functional principal components of each group (coefficients $(\lambda_{i1}, \lambda_{i2})$ for the ith curve, as defined in Section 2.2). These figures provide useful and interpretable maps of the temperature functions. For instance, the first axis of each subspace seems to discriminate the North and South cities. The figures also highlight the similarity between the temperatures of Atlantic and Pacific stations. It also appears that, in this case, the four functional subspaces seem to be parallel (same orientations but different means). To summarize, this first experiment has highlighted that funHDDC, in addition to providing a meaningful partition of the data, allows interpretations which would be certainly helpful in many application fields.

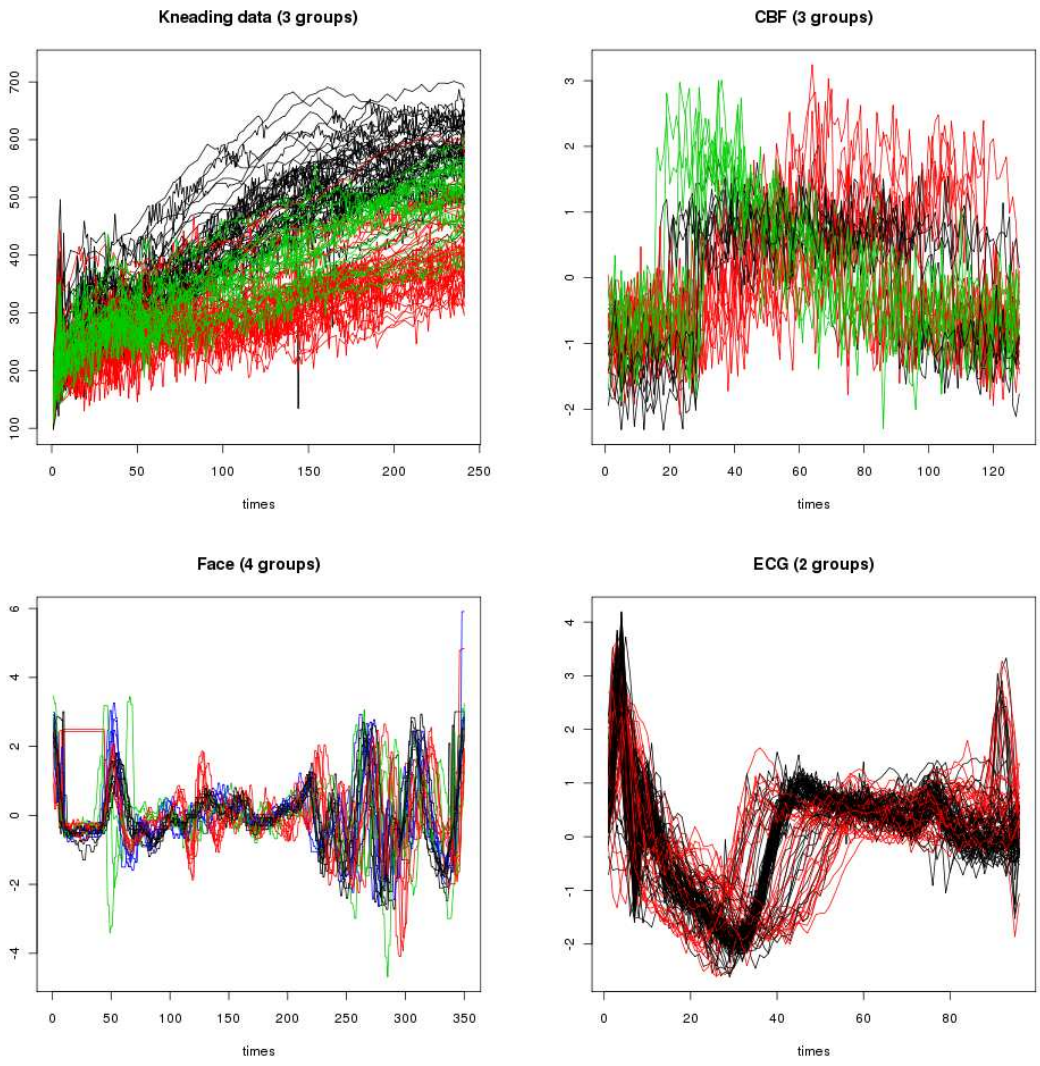


Figure 8: *Kneading*, *CBF*, *Face* and *ECG* datasets.

4.2 Benchmark study: data and experimental setup

In the two following benchmark experiments, four real datasets will be under study: the *Kneading*, *CBF*, *Face* and *ECG* datasets. These four datasets are plotted on Figure 8. The first dataset (*Kneading*) comes from a study which aimed to predict the quality of cookies (good, adjustable or bad) from the kneading curve representing the resistance (density) of dough observed during the kneading process. The corresponding dataset is made of 115 curves observed at 241 equi-spaced time points. Among the 115 cookies, 50 have been rated as good, 25 adjustable and 40 bad. These data, provided by the Danone company, have been already studied in a supervised classification context (Lévéder et al, 2004; Preda et al, 2007). These data are known to be hard to discriminate, even for supervised classifiers, partly because of the adjustable class. The three other datasets are taken from the *UCR Time Se-*

ries Classification and Clustering website[1]. The *CBF* dataset is made of 930 curves sampled from 3 groups at 128 instants of time. The *Face* dataset (Xi et al, 2006) consists of 112 curves sampled from 4 groups at 350 instants of time. Finally, the *ECG* dataset (Olszewski, 2001) consists of 200 curves from 2 groups sampled at 96 time instants.

We also have, for each dataset, the labels indicating the group membership for each observation. These labels have been provided by human experts and are available with the data. In order to compare the clustering ability of the studied methods, we chose to use the correct classification rate (CCR) which measures the adequation of the resulting clustering with the partition provided by the experts. This measure varies between 0 and 1, and larger the CCR is, better the clustering algorithm performs.

In the following, two benchmark experiments will allow to compare the clustering ability of the funHDDC method with state-of-the-art methods. Firstly, funHDDC will be compared to the *fclust* method of James and Sugar, described in Section 1, which also takes into account the functional nature of the data. Secondly, funHDDC will be compared to usual two-step methods in which the functional data are first transformed into a finite dimensional vector (simple time discretization, projection in a natural cubic spline basis with 20 basis functions or in functional principal components basis) and then clustered by an usual clustering method (HDDC, Bouveyron et al (2007), MixtPPCA (Tipping and Bishop, 1999), GMM (Banfield and Raftery, 1993; Celeux and Govaert, 1995) through the **R** package *mclust*, k-means and hierarchical clustering *via* the **R** package *hclust*).

4.3 Benchmark study: comparison with *fclust*

A package implementing *fclust* for the **R** software is available on James's website. However, because of a memory limitation in the *fclust* package, we had to select a reduced number of curves from the original four datasets. For the Kneading data, 50 curves have been randomly chosen in the 115 original ones, and for the three other datasets, which are separated into a training and a test sample on the UCR website (for supervised classification purpose), only the training part have been kept. For funHDDC, a basis of 20 natural cubic splines has been chosen for each dataset. The clustering results are provided by Table 2 which indicates the correct classification rates for both methods, the BIC values and the intrinsic dimensions for each group-specific functional subspace for funHDDC. These results clearly show that funHDDC outperforms *fclust* on all the datasets. Moreover, it appears that the BIC criterion, used for choosing the number of dimensions (tuned by a common threshold) and the most appropriate submodel, often leads to select the most efficient funHDDC models according to the highest correct classification rate (for three datasets among four). It should nevertheless be noticed that *fclust* has been developed especially for sparsely sampled functional data, and it would be interesting to compare both methods on such data too.

[1] *http://www.cs.ucr.edu/∼eamonn/time_series_data/*

Dataset	Kneading			CBF		
Number of groups	3			3		
Size	50			30		
Method	CCR	BIC	d	CCR	BIC	d
Fun-HDDC $[a_{kj}b_kQ_kd_k]$	70	-2403	(2,1,1)	63.3	-2430	(1,1,1)
Fun-HDDC $[a_{kj}bQ_kd_k]$	66.6	-2498	(1,1,1)	63.3	-2498	(1,1,1)
Fun-HDDC $[a_kb_kQ_kd_k]$	**70**	**-2193**	(1,1,1)	56.6	-2514	(1,1,1)
Fun-HDDC $[a_kbQ_kd_k]$	66.6	-2402	(1,1,1)	63.3	-2402	(1,1,1)
Fun-HDDC $[ab_kQ_kd_k]$	66.6	-2195	(1,2,1)	56.6	-2523	(1,1,1)
Fun-HDDC $[abQ_kd_k]$	66.6	-2397	(1,1,1)	**63.3**	**-2397**	(1,1,1)
fclust	60			56.6		

Dataset	Face			ECG		
Number of groups	4			2		
Size	24			100		
Method	CCR	BIC	d	CCR	BIC	d
Fun-HDDC $[a_{kj}b_kQ_kd_k]$	62.5	-2162	(1,1,2,1)	77	-6667	(1,1)
Fun-HDDC $[a_{kj}bQ_kd_k]$	50	-2286	1,1,1,1)	76	-6428	(1,1)
Fun-HDDC $[a_kb_kQ_kd_k]$	**62.5**	**-2078**	(2,1,1,1)	77	-6333	(1,1)
Fun-HDDC $[a_kbQ_kd_k]$	58.3	-2083	(1,2,1,1)	77	-6191	(1,1)
Fun-HDDC $[ab_kQ_kd_k]$	66.6	-2092	(2,1,2,1)	77	-6317	(1,1)
Fun-HDDC $[abQ_kd_k]$	58.3	-2080	(2,1,1,1)	**77**	**-6167**	(1,1)
fclust	41.6			75		

Table 2: Correct classification rates (CCR) in percentage, BIC values (if available), and dimension of each class-specific functional subspace (d) for methods *fclust* and funHDDC on parts of the Kneading, CBF, Face and ECG datasets.

Fun-HDDC	Kneading	2-steps	Kneading		
	functional	methods	discretized (241 instants)	spline coeff. (20 splines)	FPCA scores (4 components)
$[a_{kj}b_kQ_kd_k]$	64.35	HDDC	**66.09**	53.91	44.35
$[a_{kj}bQ_kd_k]$	62.61	MixtPPCA	65.22	64.35	62.61
$[a_kb_kQ_kd_k]$	64.35	mclust	63.48	50.43	60
$[a_kbQ_kd_k]$	62.61	k-means	62.61	62.61	62.61
$[ab_kQ_kd_k]$	64.35	hclust	63.48	63.48	63.48
$[abQ_kd_k]$	62.61				

Fun-HDDC	CBF	2-steps	CBF		
	functional	methods	discretized (128 instants)	spline coeff. (20 splines)	FPCA scores (17 components)
$[a_{kj}b_kQ_kd_k]$	64.84	HDDC	68.60	51.18	68.17
$[a_{kj}bQ_kd_k]$	70.43	MixtPPCA	65.59	51.29	68.27
$[a_kb_kQ_kd_k]$	64.09	mclust	61.18	62.79	68.06
$[a_kbQ_kd_k]$	**70.65**	k-means	64.95	54.09	64.84
$[ab_kQ_kd_k]$	70.65	hclust	60.86	57.96	66.13
$[abQ_kd_k]$	70.65				

Fun-HDDC	Face	2-steps	Face		
	functional	methods	discretized (350 instants)	spline coeff. (20 splines)	FPCA scores (3 components)
$[a_{kj}b_kQ_kd_k]$	56.25	HDDC	59.82	58.03	63.39
$[a_{kj}bQ_kd_k]$	54.44	MixtPPCA	54.54	61.36	**64.77**
$[a_kb_kQ_kd_k]$	51.78	mclust	62.5	57.14	55.36
$[a_kbQ_kd_k]$	54.44	k-means	59.09	53.41	59.09
$[ab_kQ_kd_k]$	60.71	hclust	50.89	56.25	48.21
$[abQ_kd_k]$	57.14				

Fun-HDDC	ECG	2-steps	ECG		
	functional	methods	discretized (96 instants)	spline coeff. (20 splines)	FPCA scores (19 components)
$[a_{kj}b_kQ_kd_k]$	75	HDDC	74.5	73.5	74.5
$[a_{kj}bQ_kd_k]$	-	MixtPPCA	74.5	73.5	74.5
$[a_kb_kQ_kd_k]$	76.5	mclust	81	80.5	**81.5**
$[a_kbQ_kd_k]$	74.5	k-means	74.5	72.5	74.5
$[ab_kQ_kd_k]$	76.5	hclust	73	76.5	64
$[abQ_kd_k]$	75				

Table 3: Correct classification rates (CCR) in percentage for funHDDC (underlined for the best model according BIC) and usual two-step methods on the Kneading, CBF, Face and ECG datasets.

4.4 Benchmark study: comparison with usual two-step methods

In this section, the clustering performance of funHDDC is compared to the usual two-step methods described in Section 1. The clustering results are summarized in Table 3. For the four datasets, the correct classification rates of each funHDDC submodel is provided, as well as for five classical clustering methods: HDDC, MixtPPCA, *mclust*, k-means and *hclust*. All these two-step methods are successively applied on discretized data, on the coefficients in a natural cubic splines basis expansion (20 splines) and on functional PCA scores. For funHDDC, also applied with a basis of 20 natural cubic splines, the correct classification of the best model according to BIC is underlined.

In view of the results of Table 3, we can make two important remarks. Firstly, funHDDC appears to outperform HDDC on the four datasets when HDDC is applied on spline coefficients. This demonstrates that taking into account the functional nature of the data in the model of funHDDC allows to improve the clustering

results compared to HDDC. Secondly, this benchmark study highlights the difficulty of the discretization choice for the two-step methods. Indeed, each of the studied method, except k-means, turned out to be the best method at least once over the four datasets and this with a different discretization choice each time. In addition, since the corresponding space in which the functions are represented are not similar, model selection criteria cannot be used to choose between the discretization strategies in an unsupervised classification context. From this point of view, funHDDC appears to be a good alternative to two-step clustering methods for the clustering of functional data since funHDDC presents the advantage of always providing satisfying results in addition to not requiring to transform the functional data into finite dimensional data. The use of funHDDC appears overall to be more tenable than two-step methods, since the funHDDC submodel selected by BIC leads to a satisfying classification rate for each dataset.

5 Conclusion

The main objective of the present work was to adapt the HDDC clustering method to functional data. The resulting algorithm, called funHDDC, models and clusters the high-dimensional functional data of each group in a specific functional subspace. The clustering and interpretation abilities of funHDDC have been illustrated on several real-world datasets. In particular, funHDDC has been applied to the well-known Canadian temperature dataset and it provided meaningful and understandable results. The proposed method has also been compared on four benchmark datasets with a recent functional clustering method, *fclust*, and with classical two-step methods. On the one hand, funHDDC turned out to clearly outperforms its functional challenger *fclust*. On the other hand, funHDDC appeared to be always satisfying and more stable than the two-step methods which furthermore suffer from the difficulty to choose the discretization strategy. We want also to mention that we considered only a cubic spline basis in this paper, but other basis functions like P-splines or wavelets could bring good results. An extension of this work would be to adapt the funHDDC method to multi-dimensional time series. This would be possible by using a Gaussian model with block-diagonal covariance matrices within the group-specific functional subspaces.

Acknowledgements

We would like to thank Prof. Cristian Preda for numerous comments and discussions that contributed to this work. We would like also to thank the three reviewers and the editor whose comments and suggestions greatly helped to improve the presentation of this paper.

References

Aguilera A, Escabiasa M, Preda C, Saporta G (2011) Using basis expansions for estimating functional PLS regression. applications with chemometric data. Chemometrics and Intelligent Laboratory Systems 104(2):289–305

Banfield J, Raftery A (1993) Model-based Gaussian and non-Gaussian clustering. Biometrics 49:803–821

Biernacki C (2004) Initializing EM using the properties of its trajectories in Gaussian mixtures. Statistics and Computing 14(3):267–279

Bouveyron C, Girard S, Schmid C (2007) High Dimensional Data Clustering. Computational Statistics and Data Analysis 52:502–519

Cattell R (1966) The scree test for the number of factors. Multivariate Behav Res 1(2):245–276

Celeux G, Govaert G (1995) Gaussian parsimonious clustering models. The Journal of the Pattern Recognition Society 28:781–793

Delaigle A, Hall P (2010) Defining probability density for a distribution of random functions. The Annals of Statistics 38:1171–1193

Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39(1):1–38

Escabias M, Aguilera A, Valderrama M (2005) Modeling environmental data by functional principal component logistic regression. Environmetrics 16:95–107

Ferraty F, Vieu P (2006) Nonparametric functional data analysis. Springer Series in Statistics, Springer, New York

Frühwirth-Schnatter S, Kaufmann S (2008) Model-based clustering of multiple time series. Journal of Business and Economic Statistics 26:78–89

Hartigan J, Wong M (1978) Algorithm as 1326 : A k-means clustering algorithm. Applied Statistics 28:100–108

Jacques J, Bouveyron C, Girard S, Devos O, Duponchel L, , Ruckebusch C (2010) Gaussian mixture models for the classification of high-dimensional vibrational spectroscopy data. Journal of Chemometrics 24:719–727

James G, Sugar C (2003) Clustering for sparsely sampled functional data. J Amer Statist Assoc 98(462):397–408

Lévéder C, Abraham P, Cornillon E, Matzner-Lober E, Molinari N (2004) Discrimination de courbes de prétrissage. In: Chimiométrie 2004, Paris, pp 37–43

Olszewski R (2001) Generalized feature extraction for structural pattern recognition in time-series data. PhD thesis, Carnegie Mellon University, Pittsburgh, PA

Preda C, Saporta G, Lévéder C (2007) PLS classification of functional data. Comput Statist 22(2):223–235

Ramsay JO, Silverman BW (2005) Functional data analysis, 2nd edn. Springer Series in Statistics, Springer, New York

Schwarz G (1978) Estimating the dimension of a model. Ann Statist 6:461–464

Tarpey T, Kinateder K (2003) Clustering functional data. J Classification 20(1):93–114

Tipping ME, Bishop C (1999) Mixtures of principal component analyzers. Neural Computation 11(2):443–482

Wahba G (1990) Spline models for observational data. SIAM, Philadelphia

Warren Liao T (2005) Clustering of time series data – a survey. Pattern Recognition 38:1857–1874

Xi X, Keogh E, Shelton C, Wei L, Ratanamahatana C (2006) Fast time series classification using numerosity reduction. In: 23rd International Conference on Machine Learning (ICML 2006), Pittsburgh, PA, pp 1033–1040